Supreme Court, U. S.
FILED
SEP 3 1975
MICHAEL RODAK, JR., CLERK

IN THE
SUPREME COURT OF THE UNITED STATES
OCTOBER TERM 1975

No. 74-1033

C. Marshall Dann,
Commissioner of Patents & Trademarks,
Petitioner,

—v.—

Thomas R. Johnston,
Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES
COURT OF CUSTOMS AND PATENT APPEALS

BRIEF *AMICUS CURIAE* FOR UNIVERSAL SOFTWARE, INC.

Charles Winn Sims
2208 Piedmont
Wichita Falls, Texas 76308
Counsel for Amicus Curiae

Of Counsel:

Francis Noel Carten
Carten & Sims
500 Summer Street
Stamford, Connecticut 06901

32 Broadway
Suite 514
New York, New York 10004

ERRATA

Page 1, in the caption, change "Patients" to "Patents".

Page 8, line 5, change "relationship" to "relationships".

Page 9, footnote 7, line 21, change "correspondense" to "correspondence".

Page 10, line 27, change "mis:apprehension" to "misapprehension".

Page 22, footnote 19, line 8, before "C24-5036-3" insert "Form".

Page 25, line 4, change "equivalence" to "equivalent".

Page 25, line 28, change "contraindicted" to "contraindicated".

Page 29, line 6, change "p. 18" to "p. 16".

Page 34, line 28, before "process" insert "a".

TABLE OF CONTENTS

		Page
I.	Questions Presented	1
II.	Interest of Universal Software, Inc.	2
III.	The Nature of the Computer and Computer Programs	3
IV.	Argument	8
	A. The Lower Court Correctly Refrained From Applying *Gottschalk v. Benson*	8
	B. The Inventive Process, The Persons Skilled In The Art, And The Technological Realities Of Program Machine Design Must Be Understood In Order To Assess Inventions In This Art	13
	1. The Inventive Process In The Art Of Program Machine Design	13
	2. Persons Skilled In The Art Of Program Machine Design	14
	3. The Interaction Of The Empiricist, The Systems Architect, And The Academician	16
	4. An Explanatory Paradigm	19
	5. The Technological Equivalence Of Hardware, Firmware, And Software	22
	6. Higher Level Languages As Definitions Of Machines Defining Alternative Modes Of Implementation	26
	7. Summary	30
	C. An Unobvious Idea Can Form A Sound Legal Basis For Patentability Even Though Its Implementation Is Obvious Once The Idea Has Been Grasped	31
V.	Conclusion	36

TABLE OF AUTHORITIES

Cases: Page

Anderson's Black Rock v. Pavement Salvage Co., Inc., 396 U.S. 57 (1969) 12

Application of Abrams, 188 F.2d 165 (C.C.P.A. 1951) 14

Application of Antonson, 272 F.2d 948 (C.C.P.A. 1959) 32

Application of Bencker, 96 F.2d 326 (C.C.P.A. 1938) 32

Application of Benson, 441 F.2d 682 (C.C.P.A. 1971), reversed on other grounds *sub nom. Gottschalk v. Benson*, 409 U.S. 63 14

Application of Bernhart, 417 F.2d 1395 (C.C.P.A. 1969) 14

Application of Bisley, 197 F.2d 355 (C.C.P.A.1952) 32

Application of Christensen, 478 F.2d 1392 (C.C.P.A. 1973) 14

Application of Conover, 304 F.2d 680 (C.C.P.A. 1962) 32

Application of Deakins, 96 F.2d 845 (C.C.P.A. 1938) 32

Application of DeLancey, 159 F.2d 737 (C.C.P.A. 1947) 32

Application of Earle, 102 F.2d 232 (C.C.P.A. 1939) 32

Application of Emmey, 161 F.2d 754 (C.C.P.A. 1947) 32

Application of Foster, 438 F.2d 1011 (C.C.P.A. 1971) 14

Application of Hamilton, 64 F.2d 141 (C.C.P.A. 1933) 32

Application of Heritage, 150 F.2d 554 (C.C.P.A. 1945) 14

Application of Hofmann, 95 F.2d 257 (C.C.P.A. 1938) 32

Application of Holt, 162 F.2d 472 (C.C.P.A. 1947) 32

Application of Hortman, 264 F.2d 911 (C.C.P.A. 1959) 32

Application of Jones, 373 F.2d 1007 (C.C.P.A. 1967) 14

Application of Linnert, 309 F.2d 498 (C.C.P.A. 1962) 32

Application of Lundberg, 197 F.2d 336 (C.C.P.A. 1952) 14

Application of Mahoney, 412 F.2d 742 (C.C.P.A. 1970) 14

Application of McIlroy, 442 F.2d 1397 (C.C.P.A. 1971) 14

Application of Musgrave, 431 F.2d 882 (C.C.P.A. 1970) 14

Application of Nomiya, 509 F.2d 566 (C.C.P.A. 1975) 32

Application of Pennington, 241 F.2d 750 (C.C.P.A. 1957) 32

Application of Prater, 415 F.2d 1378 (C.C.P.A. 1968), mod. on reh. 415 F.2d 1393 (C.C.P.A. 1969) 14

Application of Roberts, 470 F.2d 1399 (C.C.P.A. 1973) 32

Application of Shaffer, 229 F.2d 476 (C.C.P.A. 1956) 32

Application of Sponnoble, 405 F.2d 578 (C.C.P.A. 1969) 32

Application of Tanczyn, 202 F.2d 785 (C.C.P.A. 1953) 32

Application of Venner, 262 F.2d 91 (C.C.P.A. 1958) 14

Application of Waldbaum, 457 F.2d 997 (C.C.P.A. 1972) 14

Armour Pharmaceutical Co. v. Richardson-Merrell, Inc., 396 F.2d 70 (3rd Cir. 1968) 34,35

The Davison Chemical Corp. v. Joliet Chemicals, Inc., 179 F.2d 793 (7th Cir. 1950) 34,35

Eibel Process Company v. Minnesota & Ontario Paper Company, 261 U.S. 45 (1923) 32

Ex parte Breidt, 161 U.S.P.Q. 767 (P.O. Bd. App. 1968) 32

Ex parte Denny, 65 U.S.P.Q. 523 (P.O. Bd. App. 1944) 32

Ex parte Greider, 17 U.S.P.Q. 73 (P.O. Bd. App. 1932) 32

Ex parte Harr, 73 U.S.P.Q. 120 (P.O. Bd. App. 1946) 32

Ex parte Hemig, 125 U.S.P.Q. 75 (P.O. Bd. App. 1959) 32

Ex parte Ligh, 159 U.S.P.Q. 61 (P.O. Bd. App. 1967) 32

Ex parte Phair, 1 U.S.P.Q. 133 (P.O. Bd. App. 1929) 32

Funk Bros. Seed Co. v. Kalo Inoculant Co., 333 U.S. 127 (1948) 33,35

General Electric Company v. Hartman, 187 Fed. 131 (1st Cir. 1911) 32

Gottschalk v. Benson, 409 U.S. 63 (1972) ..... passim

Halliburton Oil Well Cementing Co. v. Walker, 146 F.2d 817 (9th Cir. 1944) 14

Hobbs v. Beach, 180 U.S. 383 (1901) 32

Loew's Drive-In Theatres, Inc. v. Park-In Theatres, Inc., 174 F.2d 547 (1st Cir. 1949), cert. denied 338 U.S. 822 35

National Lead Co. v. Western Lead Products Co., 324 F.2d 539 (9th Cir. 1963) 34,35

Rosemary Mfg. Co. v. Halifax Cotton Mills, 257 Fed. 321 (4th Cir. 1919) 33

Trio Process Corp. v. L. Goldstein's Sons, Inc., 461 F.2d 66 (3rd Cir. 1972) 32

Statutes and Rules:

U.S. Sup. Ct. Rule 42(2), 28 U.S.C. 2
35 U.S.C. §100 (b) 8
35 U.S.C. §101 1,2,10
35 U.S.C. §102 30
35 U.S.C. §103 30
35 U.S.C. §112 14

Miscellaneous:

J. Allen, "Computer Architecture For Signal Processing", *Proceedings of the IEEE*, Vol. 63, No. 4 (April 1975) 25

B.W. Arden, B.A. Galler, T.C. O'Brian and F.H. Westervelt, "Program and Addressing Structure in a Time-Sharing Environment", *Journal ACM*, Vol. 13, No. 1 (January 1966) 19

G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*, The MacMillan Company, New York (1958) 9

N. Chomsky and M.P. Schutzenberger, "The Algebraic Theory of Context-Free Languages", *Computer Programming and Formal Systems*, North-Holland, Amsterdam (1963) 27

A. Church, "The Calculi of Lamda-Conversion", *Annals of Mathematics Studies*, Princeton (1941) .. 30

Computerworld, Vol. IX, No. 33 (August 13, 1975) 31

M. Davis, *The Undecidable*, Raven Press, Hewlitt, N. Y. (1965) 30

J.B. Dennis, "Segmentation and the Design of Multiprogrammed Computer Systems", *Journal ACM*, Vol. 12, No. 4 (October 1965) 19

W.H. Desmonde, *Real-Time Data Processing Systems: Introductory Concepts*, Prentice-Hall, Inc. New Jersey (1964) 19

C.A. Desoer and E.S. Kuh, *Basic Circuit Theory*, McGraw Hill Book Company (1966) 14

E.W. Dijkstra, *Cooperating Sequential Processes* (Preliminary Version), Mathematics Department, Technological University, Eindhoven, The Netherlands (September 1965) 19

E.W. Dijkstra, "Solution of a Problem in Concurrent Programming Control", *Comm. ACM*, Vol. 8, No. 9 (September 1965) 19

E.W. Dijkstra, "The Structure of the T.H.E. Multiprogramming System", *Comm. ACM*, Vol. 11, No. 5 (May 1968) 17,20

J. Eichel, M.Paul, F.L. Bauer and K. Samelson, "A Syntax-Controlled Generator of Formal Language Processors", *Comm. ACM*, Vol. 6, 8 (Aug. 1963) 27

R.W. Floyd, "Syntactic Analysis and Operator Precedence", *Journal ACM*, Vol. 10, 3 (July 1963) 27

U.O. Gagliardi, "Trends In Computing System Architecture", *Proceedings of the IEEE*, Vol. 63, No. 6 (June 1975) 17,22

IBM System/360 Disk Operating System, Data Management Concepts, Form C24-3427 22

IBM System/360 Disk Operating System Supervisor and Input/Output Macros, Form C24-5037 22

IBM System/360 Disk Operating System, System Control and Systems Service Programs, C24-5036-3 22

IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, Form C24-3414 22

IBM System/360 Principles of Operation, Form A22-6821 22

E.T. Irons, "Structural Connections in Formal Languages", *Comm. ACM*, Vol. 7, 2 (Feb. 1964) . 27

S.C. Kleen, "General Recursive Functions of Natural Numbers," *Math Annalen*, (1936) 30

D.W. Knuth, *Comm. ACM*, Vol. 9, No. 5 (May 1966) 19

B. Krohn and J.L. Rhodes, "Algebraic Theory of Machines," *Mathematical Theory of Automata*, Polytechnic Press Brooklyn (1963) 9

S.E. Madnick and J.J. Donovan, *Operating Systems*, McGraw-Hill Book Company, New York (1974) 17

Master Control Program Characteristics, B5500 Information Processing Systems Bulletin 5000-21003, Burroughs Corp. (May 1962) 19

M.L. Minsky, *Computation: Finite And Infinite State Machines*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1967) 4,29

A.S. Noble, Jr. "Design of An Integrated Programming and Operating System. Part I. System Considerations and the Monitor", *IBM Systems Journal*, Vol. 7, No. 2 (June 1963) 19

Olmsted, *Intermediate Analysis-Theory of Functions Of One Real Variable*, Appleton-Century-Crofts, New York, N.Y. (1956) 7

E.L. Post, "Absolutely Unsolvable Problems And Relatively Undecidable Propositions — Account Of An Anticipation", M.S. unpublished, 1941, formally published, M. Davis, *The Undecidable*, Raven Press, Hewlett, N.Y. (1965) 30

E. L. Post, "A Variant Of A Recursively Unsolvable Problem", *Bull. Amer. Math. Soc.*, Vol. 52 (1946) 30

O. Rabin and D. Scott, "Finite Automata And Their Decision Problems," *IBM Journal of Research and Development*, Vol. 3, No. 2 (April 1959) 9

B. Randall and L.J. Russell, *ALGOL 60 Implementation*, Academic Press, New York (1964) 27

The Random House Dictionary of the English Language (Unabridged Edition) (1966) 31

J. Scafetta, Jr., "Computer Software Protection: The Copyright Revision Bills and Alternatives", *The John Marshall Journal of Practice & Procedure*, Vol. 8, No. 3 (Spring Term, 1975) 24,31

M.P. Schutzenberger, "On Finite Monoids Having Only Trivial Subgroups", *Information and Control* 8 (1965) 9

A.M. Turing, "On Computable Numbers, With An Application to the Entscheidungsproblem," *Proc. London Math Soc.*, Ser. 2-42 (1936) 7,30

U.S. Patent No. 3,500,062 24

H. Wang, "A Variant to Turing's Theory Of Computing Machines", *Journal ACM*, Vol. 4, No. 1 (1957) 7

Webster's New World Dictionary of The American Language, Second College Edition (1972) 27

N. Wirth, "Find Precedence Functions Algorithm 265", *Comm. ACM*, Vol. 8, 10 (Oct. 1965) 27

N. Wirth, A Note on "Program Structures for Parallel Processing" *Comm. ACM*, Vol. 9, No. 5 (May 1966) 19

N. Wirth, H. Weber, "Euler — A Generalization of ALGOL And Its Formal Definition." Part I, Part II., *Comm. ACM*, Vol. 9, 1 (Jan. 1966) and Vol. 9,2 (Feb. 1966) 27

IN THE SUPREME COURT OF THE UNITED STATES

OCTOBER TERM 1975

No. 74-1033

C. Marshall Dann,
Commissioner of Patients & Trademarks,
Petitioner,

—v.—

Thomas R. Johnston,
Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES
COURT OF CUSTOMS AND PATENT APPEALS

BRIEF *AMICUS CURIAE*
FOR
UNIVERSAL SOFTWARE, INC.

I

QUESTIONS PRESENTED

1. Whether a new machine, which is defined as patentable subject matter under 35 USC §101, becomes unpatentable if it is formed by the interconnection of components of a so-called general purpose digital computer which is known, the interconnections being effected by novel and unobvious program software to define the new machine's specification.

2. Whether the collection of structurally and functionally interrelated components commonly referred to as a gen-

eral purpose digital computer, when restructured by program software to form a physical realization of a machine specification of an accounting machine, comprises patentable subject matter under 35 USC §101 when accurately claimed as a machine system.

II

INTEREST OF UNIVERSAL SOFTWARE, INC.

This brief *amicus curiae* is submitted by Universal Software, Inc., a Connecticut corporation, (hereinafter referred to as "Universal") in accordance with Rule 42(2) of this Court. Both petitioner and respondent have consented to the filing of this brief. The letters of consent have been filed with the Clerk of this Court, accompanying the required number of copies of this brief.

Universal is engaged in the design, manufacture, and sale of machine systems specifications (program software) which, when combined with the hardware commonly known as general purpose digital computers, restructures such hardware into new and enhanced machines having capabilities not present in the hardware as sold by the manufacturers. Universal is exemplary of independent software houses who produce and sell machine specifications of either enhanced general purpose digital computers or special purpose digital computers. These software producers have made heavy investments of their ingenuity, time, effort, and capital in making highly significant contributions to the arts of computer systems design, which contributions are deserving of patent protection.

Although Universal does not have any direct interest in the patent application of respondent, Universal does have a deep and abiding interest in the determination of the broader issues in this case, since this Court's decision presumably will resolve the question of patentability of computer program software in a machine system claim format. The resolution of these issues in favor of respondent will affect the development of the data processing industry in the United States in a beneficial manner, encouraging disclosure of commercially viable and technically superior products to potential investors in the financial community and to potential users and improvers in the technical community, whose interests are best served by making available all existing knowledge of the state of the art. Such disclosures may be in the form of either a patent specification, or a purely defensive publication aimed at foreclosing the grant to another of a patent on the published invention. Thus, the widest possible dissemination of knowledge of the art will result, and thereby insure that advances will occur in the program software arts at an accelerated rate characteristic of developments in other crucial and dynamic technologies. The ultimate effect of a holding that computer programs of the type sought to be patented in this case are patentable subject matter would be to encourage competition, not to foreclose it.

This brief undertakes to refute certain inappropriate technical characterizations of program inventions, which, regrettably, have gained currency and caused great uncertainty in the law. Alternative characterizations more in keeping with technical reality will be presented, showing that a program invention such as that of respondent (1) is statutory subject matter and (2) is appropriately claimed as a machine system, the manner best suited to set forth the invention in its true dimensions.

Thus, it is Universal's position that the majority of the Lower Court correctly refrained from applying this Court's decision in *Gottschalk v. Benson*, 409 U.S. 63 (1972).

III

THE NATURE OF THE COMPUTER AND COMPUTER PROGRAMS

The machine known as a general purpose digital computer affords a tremendous problem-solving capability by providing within a relatively small space all of the various components requisite for fashioning more powerful machines for solving problems derivative from a highly-industrialized environment. Circuits are uniquely

defined by program software interconnection of the various components, these circuits being suitable for storing, retrieving, organizing, and manipulating signals representative of parameters meaningful within the field of interest. Thus, the computer expands the ability of the technical community to translate problems perceived in any segment of an industrial environment into machines or machine systems capable of solving such problems. Although a general purpose digital computer has the inherent potential to be fashioned into a large variety of machines capable of solving a large variety of problems, the number of machines which may be fashioned is limited by the ingenuity, time, and funds available to implement a machine solution to a particular machine problem to be solved.

Every special purpose digital computer operates solely in accordance with its specification. Its specification is its program. The program defines the various circuits which manipulate the data signals in accordance with an *effective procedure* for machine control which is accomplished in a step-by-step manner. Although the terms have frequently been confused, the effective procedure is distinct from the algorithm defining the industrial problem to be solved.[1] In this industrial context, the algorithm is the original procedure proposed to describe the conceptualization of the machine problem to be solved, expressed in a mathematical or other symbolic manner. The effective procedure is that sequence of operations which is, in fact, effective as a control mechanism and which, in fact, defines a machine solution to a machine problem to be solved.[2]

A machine specification may be created for defining a machine suitable for determining the proper course to the moon, for example. Using a convenient language such as mathematics, there is developed a model of the

[1]M.L. Minsky, *Computation: Finite And Infinite State Machines*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1967), pp. 105-107. This is presently a basic text of computer science, and was originally used by Professor Minsky as a textbook in the Department of Electrical Engineering at the Massachusetts Institute of Technology.

[2]Ibid.

environment within which the fully-formed, thoroughly-tested machine is to function. Logical sequences of operation of that machine may then be designed symbolically, for example, by a flow chart. The next step in the machine's design is the development of program software embodying a representation of a detailed set of operations for altering the general purpose digital computer's functioning to define the procedure then thought to be effective for simulating the new computer sought to be specified. This sequence of operations, or untested program, is successively refined until the new machine is completely specified. In devising the machine specification, it is necessary for the designer to be knowledgeable about the particular computer's characteristics, i.e., how it actually manipulates signals representative of the data of interest, in order for him to optimize his new machine with respect to the characteristics of the old.

Typically, the general purpose digital computer provided with the untested program is an inoperative device. Considerable skill may be required in carrying out the implementation of the machine solution, particularly in view of the limited number of fixed operations which may be performed by the general purpose digital computer. Devising the finished machine, often through many intermediate debugging steps, generally requires the continuing attention of a professional whose insights may be valuable in both the original perception of the machine problem to be solved and the design of the machine to solve it. In many instances, the inventive process is so complex as to require substantial innovative effort on the part of every member of a project team, including the computer machine process specialist.

The difficulty of conceptualization and communication of this subject matter may be readily appreciated by reference to the rather strained analogies which have been employed by Amici in *Benson* and in the instant case. Notable is the player piano analogy (Pet. Br. 16, 35).[3] No player piano includes conditional logic circuitry, which is essential in forming any digital computer. Such circuitry

[3]"Pet. Br."refers to Petitioner's Brief on the merits.

enables conditional branching, i.e., selection of the next operation on the basis of the result of a completed operation. The specification of interconnections of conditional logic circuits in a general purpose digital computer is necessary in order that the machine fashioned therefrom, namely, a special purpose digital computer, be capable of changing from one functionally operative state to another responsive to events represented by internally-manipulated data signals during the course of its functioning. Although it may be refreshing to have a player piano change its tune by reforming its internal structure without external manipulation, no such instrument has ever done so.

It has been said that a computer program is equivalent to the paper music roll instructions to a player piano (Pet. Br. 35). However, the paper music roll does not restructure the player piano; it merely causes certain notes to be played in a predetermined sequence. Thus, if a program software were truly like a piano roll, it would simply cause the general purpose digital computer to generate a predefined sequence of signals in response to the signals from the program machine. This is not the manner of interaction of the program with the general purpose computer. The technological reality is that program software causes new circuits to be formed in the *general purpose machine,*[4] thereby changing it to a *special purpose*

[4]That a general purpose digital computer may accurately be denominated a machine has a sound basis in the theories of electronics and computer science. See Minsky, *supra* note 1, p. 200.

machine[5] or an *enhanced/extended general purpose machine*[6] having new problem-solving capabilities which qualitatively or quantitatively extend beyond the capabilities of the general purpose machine.

[5]A sound theoretical basis for the proposition that a program converts a general purpose digital computer (or machine) into a special purpose digital computer (or machine) is found in Minsky, *supra* note 1, p. 200. Precise definitions adequate to the statement of a theoretical basis for this phenomenon are found in section 11.2 at p. 204 and in Theorems 14.1-1 at p. 257 and 14.2-1 at p. 258. Significant milestones in development of such theoretical basis are the following papers: A.M. Turing, "On Computable Numbers, With An Application to the Entscheidungsproblem," *Proc. London Math. Soc.*, Ser. 2-42 (1936), pp. 230-265; H. Wang, "A Variant to Turing's Theory of Computing Machines", *Journal ACM*, Vol. 4, No. 1 (1957).

[6]An application program defines structure which is a special purpose digital computer. A system program defines structure which is an enhanced or extended general purpose digital computer. A straightforward mathematical technique will be applied to the consideration of the structure of machines. In the strictest sense, it is mathematical only to the extent that logic is mathematics, an important proposition in the study of machines. There will be applied the technique of inductive proof.

For every positive integer n, let P(n) be a proposition that is either true or false. If (i) P(1) is true and (ii) whenever this proposition is true for the positive integer n it is also true for the positive integer n + 1, then P(n) is true for all positive integers n.

Proposition: A new program defines a new machine.

Let an application program represent the proposition P(1). Then no system program is consistent with the Proposition, i.e., there exists no program to which an integer greater than 1 may be assigned, all system programs being available when the application program is defined. We have a contradiction, so that it is irrational to start with the application program.

Let some system program represent P(1), another P(n) and another P(n + 1). Then if the Proposition can be shown to be true, it will be true also when choosing n to represent the last system program found by the application programmer to be helpful in designing his machine, and n + 1 to represent identically the application program.

By this reasoning alone there is no structural difference between a system program and an application program.

Induction is succinctly defined in Olmsted, *Intermediate Analysis-Theory of Functions Of One Real Variable*, Appleton-Century-Crofts, New York, N.Y. (1956).

A general purpose digital computer has been said to be a warehouse of functionally unrelated parts. This is true only in the sense that the relationships of the parts are not defined for the solution of any specific problem. The relationship of the parts within a general purpose machine are well-defined in the mechanical sense, but are almost endlessly variable in the electrical sense. Program software in combination with a general purpose digital computer creates what has been termed "instant hardware" in the form of either an enhanced general purpose digital computer or a special purpose digital computer by making the electrical connections which form those computers.

IV

ARGUMENT

A. The Lower Court Correctly Refrained From Applying *Gottschalk v. Benson*

In *Gottschalk v. Benson,* this Court decided essentially that a signal conversion method which is defined in mathematical terms is not a process within the meaning of 35 U.S.C. §100(b). It does not appear to have been appreciated that mathematics has a broad, generally-accepted definition which encompasses not only number systems and the manipulation of numbers, but also logistics, axiomatic investigation, symbolic language, and, for that matter, experience. Since mathematics is the language employed by engineers and scientists to express their experience, it is unrealistic to have a definition of mathematics in law different than that understood by Euclid, Descartes, Leibniz, Gauss, LaPlace, Newton, Fourier and most significantly, Galois, the Father of

Modern Mathematics, whose notion of isomorphism[7] leads one to the conclusion that a mathematician is more nearly a pioneering observer than a chronicler. That which is best communicated in terms of a language mathematical in nature and experiential in content and context should be available to describe an invention which has been conceived and reduced to practice as a procedure or sequence of steps. In any event, it was not held that an exhaustive list of patentable processes had been compiled by the year 1972, nor are processes, de-

---

[7]That Evariste Galois is often referred to as The Father of Modern Mathematics is attributed to the extraordinary breadth of modern structures whose behavior can be examined, explained and digested with reference to his original notion of isomorphism, defined most simply as a one-to-one correspondence between elements of groups. The groups of interest have certain properties, the most important of which is the manner in which elements are combined while maintaining their identity as members of that group by the definition of membership therein. The one-to-one correspondence survives combinations. This is the basis for abstract algebra (synonym of modern algebra). G. Birkhoff and S. MacLane, *A Survey of Modern Algebra,* The MacMillan Company, New York (1958). The simplest example is the simultaneous consideration of an algebra (or set of rules) governing combinations of odd and even integers and one of the number system base two (or modulo 2) which has but two integers, 0 and 1.:

even + even = odd + odd = even	even + odd = odd
even + even = even · odd = even	odd · odd = odd
0 + 0 = 1 + 1 = 0	0 + 1 = 1
0 · 0 = 0 · 1 = 0	1 · 1 = 1

The one-one correspondense is *even* ↔ *0; odd* ↔ *1,* which is an isomorphism.

Both the most general control of instantaneous interconnections of circuits in a general purpose digital computer and the specific manner in which it counts can be explained with precisely this isomorphism. For a much deeper analysis of theories of structures of general purpose digital computers traceable to the work of Galois, there is to consider: O. Rabin and D. Scott, "Finite Automata And Their Decision Problems," *IBM Journal of Research and Development,* Vol. 3, No. 2 (April 1959), pp. 114-125; B. Krohn and J.L. Rhodes, "Algebraic Theory of Machines," *Mathematical Theory of Automata,* Polytechnic Press, Brooklyn (1963), pp. 341-384; M.P. Schutzenberger, "On Finite Monoids Having Only Trivial Subgroups", *Information and Control* 8 (1965), pp. 190-194.

Galois died in 1832 at the age of 21 years.

fined by an algorithm or otherwise, before the Court in the present case.

In the decision below in the instant case, the Court of Customs and Patent Appeals held that a general purpose digital computer combined with program software to form a machine system for performing a financial banking procedure is patentable subject matter under 35 U.S.C. §101. However, Amicus CBEMA urges that the record does not show that a machine has been invented or disclosed in respondent's patent application, and points to respondent's statements that "the invention is being sold as a program to banks" and "was reduced to practice in the software form of computer programs . . ." (CBEMA Br. 8).[8] These statements in no way conflict with respondent's assertion that such programs define a control system in the form of a special purpose digital computer. Respondent has not conceded that the merchantable product, viz., program software, is anything less than a machine for restructuring a general purpose machine to form the machine system known as a special purpose machine for carrying out a financial banking procedure. Thus, respondent's real contribution is much more than a financial banking algorithm. Since respondent's claims are drawn to a machine system, clearly the intent of respondent is to define his invention as a machine system, not as an algorithm. Amicus CBEMA's basic mis:apprehension of the commercial realities of program machine design and implementation would lead one to confuse a flow-charted and coded financial banking procedure with an effective procedure in the form of a completely debugged and tested program. Only the latter, i.e., the running program, is a commodity which is merchantable to astute and knowledgeable businessmen such as bankers and systems accountants. Absent this merchantable program machine, whether in the form of punched cards, magnetic tape or disks, the general purpose digital computer is not capable of solving any problem. Thus, a general purpose machine in combination with an application program defines a machine system for solving a problem. The notion that

[8] "CBEMA Br." refers to Amicus CBEMA's Brief on the merits.

program software *simply* defines computer routines (CBEMA Br. 8, note 9) is alien to those in the data processing community, particularly to those technically competent to carry a symbolic representation of a procedure into a running program. Any implication that simplicity is an attribute of either the design or implementation of the machine system claimed by respondent does not conform to reality.

Financial banking procedures whose implementation has been principally manual, or has involved certain automated procedures requiring frequent or continuing human intervention, have been considered methods of doing business, which like mathematics of the genus defining laws of nature or scientific principles, have been considered unpatentable *per se*. In contradistinction, respondent has disclosed and claimed a machine system, i.e., a hardware implementation of his effective procedure which is the solution to a machine problem to be solved. Therefore, respondent should be considered as having defined patentable subject matter, notwithstanding that the machine problem may have been formulated as a numerical procedure or algorithm. Unless this Court holds that respondent's application program machine is patentable subject matter as claimed, it will be impossible to adequately define the scope of novel and unobvious hardware implementations in the form best suited to convey to the technical community the essence of extremely valuable and technologically superior contributions to the art of program machines. To a person working in this art, a recitation of a particular general purpose digital computer or a class of general purpose digital computers suitable for adaptation to carry a disclosed program invention into effect in the form of a special purpose digital computer is sufficient to enable the practice of the invention disclosed in respondent's patent application, viz, a control system or special purpose digital computer. Such a disclosure would be enabling in patent applications disclosing novel and unobvious program machines of any genus, including Amicus Universal's commercially viable, novel and unobvious system program machine defining an enhanced general purpose digital computer.

Program software interactively combined with a general purpose digital computer should be considered patentable as a machine system by virtue of the synergistic interaction with the general purpose digital computer, notwithstanding that the program may be described in terms of an algorithm which may or may not be expressed in a mathematical language. *Anderson's Black Rock v. Pavement Salvage Co., Inc.*, 396 U.S. 57 (1969). In the instant case, there is precisely the synergism considered by this Court in *Anderson's*, i.e., that the program and the computer combined produce a function which neither the program nor the computer can produce alone. In any event, *Anderson's* dealt with a combination of *old elements*; the instant case deals with a *new program* in a machine system.

The interaction of the program machine with the general purpose machine is also within the definition of synergism as it is commonly understood in the biochemical sense. Such interaction is much the same as neurochemical reactions causing muscles to expand and contract in synchronism, thereby achieving the control and coordination of the musculature requisite to accomplishing a task. Such biochemical activity does not involve any intellectual process.

It is not asserted that the relation between chemical reactions in a physiological context and program control of the various components and sub-systems of the general purpose digital computer is a completely satisfactory analogy. It is simply pointed out that the simultaneous action of separate agencies toward the achieving of a given purpose is not a concept alien to human experience. However, biochemical reactions are more readily correlated with the conditional responses of a special purpose digital computer to data signals representative of an environment than the predefined responses of a player piano to the sensing of the holes in the piano roll having no more information content than some version of a composer's creation, and lacking any subtle phrasing at that.

B. The Inventive Process, The Persons Skilled In The Art, And the Technological Realities Of Program Machine Design Must be Understood In Order To Assess Inventions In This Art.

1. The Inventive Process In The Art Of Program Machine Design.

Among those making significant contributions to the art of program machine design, there is a unitary view of the process of invention of a program machine as including these elements:

(1) formulating the machine problem to be solved;

(2) flow-charting a purportedly effective procedure for solving the machine problem;

(3) devising a program machine defining said purportedly effective procedure; and

(4) implementation, i.e., getting the program onto the general purpose machine and running.

The fourth element, implementation, has not yet been treated before this Court. It is absent from the petitioner's characterization of the inventive process (Pet. Br. 16-17), notwithstanding the Pickwickian use of the word "implementation" in imprecisely defining a second "mode of activity". Petitioner has stated that it would be "useful" to consider only the first three elements of the inventive process. This narrow view is indeed "useful" if contributions to the art of program machines are to be characterized as nothing more than intellectual exercises. To this unworthy end, it is "useful" that program machines conceived and expressed originally in abstract terms be thought of as nothing more than abstractions. If this false premise is accepted, it then becomes easy to

apply the mental step doctrine[9] to deny the patentability of the enhanced general purpose machines or the special purpose machines which, like all other inventions, are the fruit of intellectual activity. Such an undesirable result is comparable to denying the patentability of electrical circuits which are the physical embodiment of a mathematical model consisting of parametric equations defining complex interrelations and interactions among circuit components.[10]

2. Persons Skilled In The Art Of Program Machine Design.

Three innovators in the art of program machine design are representative of the statutorily-hypothesized[11] "person skilled in the art". Many real-life individuals could fall within more than one of these catagories.

(1) The Empiricist

His inventions are conceived and reduced to practice, in the form of an actual implementation, on the basis of his degree of familiarity with the manufacturer's hardware, existing systems programs, and the cooperative association between the two. His machine specification, when appropriately manipulated and combined with the products of the hardware manufacturers, may be an application

[9]See *Halliburton Oil Well Cementing Co. v. Walker*, 145 F.2d 817 (9th Cir. 1944); *Application of Heritage*, 150 F.2d 554 (C.C.P.A. 1945); *Application of Abrams*, 188 F.2d 165 (C.C.P.A. 1951); *Application of Lundberg*, 197 F.2d 336 (C.C.P.A. 1952); *Application of Venner*, 262 F.2d 91 (C.C.P.A. 1958); *Application of Jones*, 373 F.2d 1007 (C.C.P.A. 1967); *Application of Prater*, 415 F.2d 1378 (C.C.P.A. 1968), mod. on reh. 415 F.2d 1393 (C.C.P.A. 1969); *Application of Bernhart*, 417 F.2d 1395 (C.C.P.A. 1969); *Application of Mahoney* 412 F.2d 742 (C.C.P.A. 1970); *Application of Musgrave* 431 F.2d 882 (C.C.P.A. 1970); *Application of Foster*, 438 F.2d 1011 (C.C.P.A. 1971); *Application of Benson*, 441 F.2d 682 (C.C.P.A. 1971), reversed on other grounds *sub nom. Gottschalk v. Benson*, 409 U.S. 63; *Application of McIlroy* 442 F.2d 1397 (C.C.P.A. 1971); *Application of Waldbaum* 457 F.2d 997 (C.C.P.A. 1972); *Application of Christensen* 478 F.2d 1392 (C.C.P.A. 1973).

[10]C.A. Desoer and E.S. Kuh, *Basic Circuit Theory*, McGraw-Hill Book Company (1966).

[11]35 U.S.C. §112, first paragraph.

program which defines a special purpose digital computer optimized with respect to the application's definition, the utilization of the hardware, and the utilization of the existing systems software. Alternatively, the Empiricist may devise a machine specification which, when appropriately manipulated and combined with the products of the hardware manufacturers, works a new synergism therebetween which is physically embodied in an enhanced or extended general purpose digital computer.

(2) The Systems Architect

His inventions are machine specifications which define the original synergism between the manufacturer's hardware and systems software, that synergism being embodied in an integrated general purpose digital computer the specification of which may define the starting point for an Empiricist concerned with implementation or development of application program machines for defining special purpose digital computers. Of course, it is also true that the original system program machine may define the starting point for an Empiricist concerned with designing an enhanced or extended general purpose digital computer. (If the second Empiricist has made his contribution, the first Empiricist would typically design his machine from the components of the enhanced machine.) The Systems Architect is an inventor, his protestations to the contrary not withstanding. His protestations are typically induced by his employers, who wish to foreclose any alternatives to the "free" systems software offered with their machines.

(3) The Academician

His contributions may be in the field of computer science, which is concerned with future implementations or machine development. The Academician has made some of his most significant contributions in the investigation of the computer and its role in society. From a purely technical standpoint, the Academicians of interest are those whose perceptions lend an insight into the nature of the computer's structure. A tool for these investigations in

many instances is mathematics of a genus which has been defined as axiomatic or logistic, as opposed to definitive and descriptive of naturally-occurring phenomena.

3. The Interaction Of The Empiricist, The Systems Architect, And The Academician.

In reality, the program machine design specialization is distributed among all three of these hypothetical innovative specialists. Thus, that one Empiricist may proceed from the work of a Systems Architect, or from the work of another Empiricist, in no way conflicts with our definitions. It is always true that the innovator must isolate his machine problem from all others, and develop his specification of a machine solution either independently or cooperatively in association with members of a project team concerned with that problem and the new machine to be fashioned for its solution.

The Systems Architect, having a keen appreciation of the principles of computer science (and engineering, to the extent that it is distinct), is able to communicate an accurate definition of the structure of the machine to the balance of the technical community, i.e., the collection of Empiricists, so that they may have a precise definition of the design parameters affecting at least the implementation portion of a specific machine problem to be solved. The Academician performs a variety of functions in advancing the state of the art. These functions may best be set forth in historical sequence of advances within two repetitive cycles. The first cycle includes Observation, Abstracting, Formalization, Propagation, Observation .... The second cycle includes Logical Formulation, Perception and Definition of Development in terms of a Formalism, Propagation, Logical Formulation .... The first cycle is relatively short in duration and more nearly responsive to immediate demands of the technical community formed by the Empiricists. There has been no completed second cycle. Certain of the demands of at least the technical community formed by the Systems Architects are met in the near term by second-cycle activities. The advancement of the state of the art of program machine design will always be the result of discrete events along the course of both cycles.

Turning to the work of the Empiricist, what exists insofar as he is concerned is precisely what he "sees". "What is seen by the application programmer or the systems programmer" is an expression frequently used in the art, principally because the structure of the general purpose digital computer is exceedingly complex. Therefore, there is a common practice of explaining its structure in terms of a paradigm[12] which facilitates communication of these labyrinthian complexities. Three working definitions of terms required to explain both the manner in which the Architect communicates his knowledge to the Empiricist and the paradigm are set forth hereunder.

Virtualization:

making available to the Empiricist an understanding of the resources of the general purpose digital computer of interest to that Empiricist as a member of a subclass of either application program machine designers or enhanced/extended machine designers.

Hierarchial view:

the viewpoint that the general purpose digital computer is best described in the technically accurate terms of machines within machines, or nested machines.[13] There are less pertinent alternative views, viz., the process view[14] and the resource management view[15], but these are merely devices facilitating understanding in a less general case. One's choice of viewpoint is determined by whether a convenient explanation or a definitive phenomenological exposition is desired. Systems Architects communicate using terminology in alignment with all three views simultaneously.[16] This

---

[12]E.W. Dijkstra, "The Structure of the T.H.E. Multiprogramming System", *Comm. ACM*, Vol. 11, No. 5 (May 1968), pp. 341-346.

[13]Ibid.

[14]S.E. Madnick and J.J. Donovan, *Operating Systems*, McGraw-Hill Book Company, New York (1974).

[15]Ibid.

[16]U.O. Gagliardi, "Trends In Computing System Architecture," *Proceedings of the IEEE*, Vol. 63, No. 6 (June 1975).

is because these views and their terminology are not in conflict as media of communication.

Machine process:

Consonant with the definitions of virtualization and the hierarchial view, a machine may be defined as a process requiring resources shared with the same or another nested machine level which may also be conveniently discussed as a process. It is convenient to think of a machine as a process because during its functioning, i.e., the performance of its unique operations, it is a *processor*. The ultimate test of its physical characterization as a machine is whether or not unique circuit interconnections within the innermost level or bare machine are defined during the time interval in which the machine exists. The term *machine process* is quite distinct from mathematical process and is, in fact, equivalent to the term *effective procedure for machine control.*

Effective communication of these highly complex concepts is more nearly contextual than semantical, and the requirement that a context be considered increases as it is determined what is to be "seen" by larger and larger numbers of Empiricists. Technically accurate alternative modes of expression are required for describing this new category of machines which may be viewed alternatively as control mechanisms, or nested machines, or machines performing operations defining machine processes which are physically realizable only through unique interconnections of circuits, thereby suggesting that the first two characterizations are the more technically accurate, and the third the more convenient. The historical

development of these modes of expression as defined herein is well-documented.[17]

4. An Explanatory Paradigm

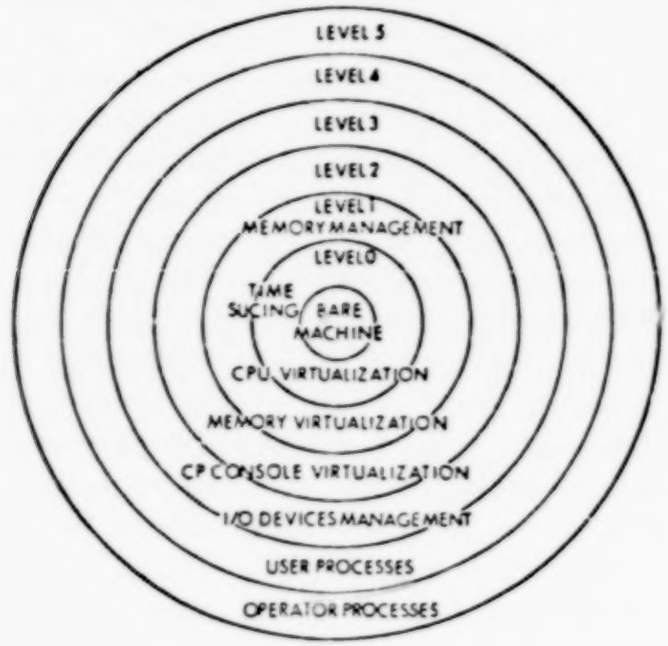

Fig. 1

---

[17]A.S. Noble, Jr. "Design of An Integrated Programming and Operating System. Part I. System Considerations and the Monitor.", *IBM Systems Journal*, Vol. 7, No. 2 (June 1963), pp. 153-161; *Master Control Program Characteristics*, B5500 Information Processing Systems Bulletin 5000-21003, Burroughs Corp. (May 1962); W.H. Desmonde, *Real-Time Data Processing Systems: Introductory Concepts*, Prentice-Hall, Inc., New Jersey (1964); E.W. Dijkstra, "Solution of a Problem in Concurrent Programming Control". *Comm. ACM*, Vol. 8, No. 9 (September 1965), p. 569; N. Wirth, A Note on "Program Structures for Parallel Processing" *Comm. ACM*, Vol. 9, No. 5 (May 1966), pp. 320-321, (letter to the editor); E.W. Dijkstra, *Cooperating Sequential Processes* (Preliminary Version), Mathematics Department, Technological University, Eindhoven, The Netherlands (September 1965); D.W. Knuth, *Comm. ACM*, Vol. 9, No. 5 (May 1966), pp. 321-322 (letter to the editor); J.B. Dennis, "Segmentation and the Design of Multiprogrammed Computer Systems", *Journal ACM*, Vol. 12, No. 4 (October 1965), pp. 589-602; B.W. Arden, B.A. Galler, T.C. O'Brian and F.H. Westervelt, "Program and Addressing Structure in a Time-Sharing Environment", *Journal ACM*, Vol. 13, No. 1 (January 1966), pp. 1-16.

The paradigm shown in Fig. 1. is an hierarchical structure[18] consonant with the hierarchial view, providing a visual conceptualization of nested machines. There is a virtualization looking from the outermost rings of the diagram inwardly, and looking from the innermost rings outwardly. By our definitions, the Empiricist is typically the one whose requirement is to have a very complex device's resources or components arranged in virtualized subsystems, while the Systems Architect, charged with the responsibility of virtualizing, would most conveniently view the program machine to be combined with his machine as a machine process (Level 4 in Fig. 1). The integrated (nested) machine with no User Process at Level 4 has no function whatever. It is a general purpose digital computer, i.e., it is only capable of being defined as a machine having a specific purpose by the machine specification of the Empiricist. No specific machine exists absent a User Process defined by the the machine specification. To the extent that the User Process has any relation to a traditional definition of process, by logical exclusion it would have to be a process for utilization of the remainder of the machine, which is equivalent to a process for sequentially defining circuit interconnections at Level 0 in Fig. 1. The aspect of the Systems Architect's work termed "looking out" addresses itself to the organization of the general purpose digital computer so that it is virtualized in an hierarchical structure facilitating the implementation phase of the Empiricist's work. A simple way of viewing this aspect is to define the innermost virtualized machine as the Central Processing Unit in combination with a facility for defining time intervals for effective procedure execution, i.e., controlling access to the various facilities consonant with the machine specification. The innermost machine facilities are thus sequentially shared by the outermost machines. This time slicing is performed by a timing mechanism for defining when the outermost machines in Fig. 1 are to have their instructions decoded in a manner defined by the Systems Architect to exercise the most general control over the innermost machine's components or subsystems in an optimum manner.

[18]Dijkstra, *supra* note 12.

In the paradigm of Fig. 1, "looking out" from Level 0 to successively higher levels is merely an organizational view of general structure that will behave in a generally predictable manner perceived by the Empiricist "looking in". To relate this to the earlier discussion, "looking out" is defining how the circuits might be interconnected as a general proposition to facilitate optimum utilization by the program machine or special purpose digital computer designer; "looking in" is determining the shallowest penetration that will be adequate to an understanding of the structure sufficient to design the new machine. Levels 0 and 1 are generally referred to as the resident nucleus. The simplest model of the resident nucleus includes (a) the Central Processing Unit, (b) the timing mechanism, (c) the main memory and (d) the most basic management of the three. The Systems Architect would normally consider the resident nucleus as exclusively within his province. The Empiricist typically designs a machine working at all levels higher than Level 1. The Systems Architect's manifold tasks include organizing and structuring the various components of the general purpose digital computer to perform optimally as the nucleus for the largest number of Empiricists desiring to structure either an enhanced general purpose digital computer or a special purpose digital computer, and communicating the organization and functioning of the nested machines to the Empiricist in a manner best suited to facilitate the accomplishment of the Empiricist's task.

Any Empiricist works with a machine of a given description to structure a new machine. Certain Empiricists would treat more with applications problems derivative from the industrial environment, and others more with systems problems defined by the functional characteristics of the original machine. This is the principal reason why the Systems Architect chooses alternative modes of

expression of the original machine's virtualization[19] to make the full capabilities or resources available to those "looking in" to any desired depth.

5. The Technological Equivalence of Hardware, Firmware, and Software.

Advances in the state of the art of machine implementation are constantly changing the boundary between the *soft* and *nonsoft* portions of the generalized computer system.[20] Computer system architecture is the collection of the various layers of basic elements or building blocks composing a computer system in their nesting relationship and with the media of implementation. The media of implementation are: (1) hardware or built-in logic, viz., the actual circuits which have the inherent capability of being interconnected to direct signals and combine them in a manner similar to the exercise of logic; (2) firmware, i.e., procedures coded in a control store affording greater speed and ease of modification than hardware, but less than software; and (3) software, i.e., procedures coded in an easily alterable main storage memory. Determining the proper media mix is a critical part of the System Architect's task, since it affects the key tradeoff, viz., ease of maintenance versus cost/performance.

---

[19]Typical examples of a Systems Architect's Virtualization of the modern general purpose digital computer to communicate its attributes to Empiricists, which examples demonstrate that by "looking in" one's perception would be that successive levels define "nested machines", are found in the following collection of one manufacturer's publications:

IBM System/360 Disk Operating System, System Control and Systems Service Programs, C24-5036-3.

IBM System/360 Disk and Tape Operating Systems, Assembler Specifications, Form C24-3414.

IBM System/360 Disk Operating System Supervisor and Input/Output Macros, Form C24-5037.

IBM System/360 Disk Operating System, Data Management Concepts, Form C24-3427.

IBM System/360 Principles of Operation, Form A22-6821.

[20]Gagliardi, *supra* note 16.

The procedures coded in the various media are effective procedures for specifically directing and combining signals in a code having the representation 1 0 1 0 1 .....meaning ON OFF ON OFF ON ..... The software medium is more easily alterable than hardware or firmware, just as it is easier to turn on the lights than to pursue the setting sun. What is embodied in the software medium is an implementation of a machine solution, designed in a manner most readily realized physically. This says nothing about its ease of design. Ease is a relative term in an engineering or scientific context, just as in a more general context. In order that anything of value be accomplished by the general purpose digital computer, thousands of switches must be turned ON or OFF during substantially the same time frame of reference. The time frame of reference is on the order of millionths or even billionths of a second. A code precisely the same as that described above is used to control the setting of the multitude of switches. These

switches reside in a plurality of circuits or arrangements of such circuits, which are defined by the setting of the switches.[21]

The running program therefore defines, in sequence, a great number of unique circuits. Thus, although program software is often called "instant hardware"[22], it is more appropriately termed "instantaneously-defined hardware combinations collectively suitable for specifying a new machine whose operational functionality is appropriate to implementing a machine solution to a machine problem to be solved." Defining the appropriate mix of implementation media may be a continuing process, even for a special purpose digital computer al-

---

[21]U.S. Patent No. 3,500,062 discloses an exemplary logic circuit which is, generically, an inverter, but provided with a plurality of elements contained therein having the inherent capability of instantaneously changing states, i.e., switches. The switch settings are eached defined logically as either O = OFF or 1 = ON at any given time, there being an additionally provided master switch which would best be defined as a *theoretical* switch, not being integrally a part of the circuit but rather being a signal serving as one of the inputs thereto. The circuit is of interest in that it is a basic building block of the general purpose digital computer, adaptable as the functional equivalent of either an EXCLUSIVE OR useful in performing arithmetic or an $\overline{\text{EXCLUSIVE OR}}$ (the logical complement) useful as a conditional logic circuit. Concerning the circuit, all of the following statements are true:

(1) Each and every switch may be set bv any of the implementation media a) hardware; b) firmware; or c) software.

(2) All of the inputs and outputs are the logical voltage equivalents of 1 or 0.

(3) The operation of the circuit is "virtualized looking out" to the extent that the source of the input signals is a matter of complete indifference.

(4) Additional circuits precisely the same as the one disclosed could be grouped almost indefinitely to define a substantial portion of the "bare machine", for all of which each of the first three statements could also be made. The output of such circuits could provide the inputs to other identical circuits, as, for example, in a multi-stage adder.

[22]J. Scafetta, Jr., "Computer Software Protection: The Copyright Revision Bills and Alternatives", *The John Marshall Journal of Practice & Procedure*, Vol. 8, No. 3 (Spring Term, 1975), p. 383.

ready designed and functioning. The three media are, as a matter of technological fact, interchangeable and therefore functionally equivalent.[23]

Exemplary of the equivalence functionality of the alternative media is Digital Signal Processing[24], a precise science requiring certain architectural insights and most readily understood by examining the environment within which the machine problem to be solved arises. The term itself may be accurately defined as directing and combining signals arriving from an external environment in such great number and at such a high rate that no matter how complex their combination, it must be accomplished at an extraordinarily high rate, as though the signals were being merely processed, a term often applied in a context wherein the combinations may be *readily* accomplished over a relatively short time frame, making time for combining signals a secondary constraint. Thus, digital signal processing machines are within the genus of *real time machines,* i.e., machines which process data almost as fast as it is received to produce correlated output signals.

Digital signal processing was originally accomplished with analog computers because of their inherent characteristic that, once a particular machine is properly designed and structured, solutions are available virtually instantaneously responsive to input signals, and because inefficiencies in operation of early models of digital computers contraindicted their choice. As the speed and efficiency of general purpose digital computers improved, the *functional equivalents of the analog machines* were designed and implemented as *special purpose digital computers,* the functionality of the digital machine being defined principally by program software. These machines yet proved to be too slow to be economical.

---

[23]J. Allen, "Computer Architecture For Signal Processing", *Proceedings of the IEEE*, Vol. 63, No. 4 (April 1975).

[24]Ibid. at p. 624: "In recent years, however, many studies have revealed structural aspects of digital signal-processing tasks which could be exploited in hardware design. As a result, there have been a number of hard-wired and programmable machines constructed which provide large speedup over general purpose processors, and which often also reduce the cost of the calculations."

As new digital technologies were developed, e.g., improvements in the switches, particularly speed-up of their inherent capability for changing states, it was observed by the Systems Architects that the critical functions might best be moved from Level 4 to Levels 0 and 1 in Fig. 1. This shifting of functions was implemented by designing new codes and a means (firmware) for their interpretation or, alternatively, by permanently wiring certain functions. Thus, all means for defining functionality in adapting a digital computer to the solution of a specific problem are interchangeable, and all such means define structure.

6. Higher Level Languages As Definitions of Machines Defining Alternative Modes of Implementation.

The development of higher level algorithm coding languages such as FORTRAN and ALGOL is typical of the work of the Academician, responding to what is called application pull, viz., the influence of real-world problems on the sequence of technical development. With the perception that the general purpose digital computer had a broad spectrum of possible end uses to which it might be adapted, there developed a requirement for making these machines more readily accessible to members of the scientific community whose interest, at an introductory or entry level of their work with these new general purpose machines, was more in translating formulae (algorithms) with relative ease than in optimizing design of the special purpose machines tailored to the solution of their problem. The intermediary between the conceptualizing individual treating principally with formulation of an algorithm in an industrial context and the particular general purpose machine sought to be adapted to that context is the Academician who abstracts the characteristics of the original machine and formulates near-optimum translation algorithms for carrying the problem formulation algorithm into effect, i.e., into an implementation or effective procedure for machine control. The intermediate procedure requisite to this mode of implementation of the machine solution within the industrial context of interest is called *compilation*,

which is accomplished under the machine control of a compiler such as FORTRAN. The result of this intermediate step, i.e., the effective procedure for machine control, rarely matches the result that obtains from the step-by-step optimum design of the special purpose machine by one more knowledgeable of the details of the general purpose machine. In short, some effective procedures are more effective than others.[25]

---

[25]The development of special purpose program machines adequate to the task of converting the higher level languages, defined simply as symbolic codes capable of expressing the algorithm as a purported effective procedure in a manner reducing greatly the steps of the coding task, has continued over a period of many years. The history of the task of expanding the codes into suboptimal step-by-step operation codes for defining circuit interconnections which are merely purported effective procedures can be traced by reference to the following:

R.W. Floyd, "Syntactic Analysis and Operator Precedence", *Journal ACM*, Vol. 10, 3 (July 1963), pp. 316-333.

J. Eichel, M. Paul, F.L. Bauer and K. Samelson, "A Syntax-Controlled Generator of Formal Language Processors", *Comm. ACM*, Vol. 6, 8 (Aug. 1963), pp. 451-455.

N. Chomsky and M.P. Schutzenberger, "The Algebraic Theory of Context-Free Languages", *Computer Programming and Formal Systems*, North-Holland, Amsterdam (1963).

B. Randall and L.J. Russell, *ALGOL 60 Implementation*, Academic Press, New York (1964).

E.T. Irons, "Structural Connections in Formal Languages", *Comm. ACM*, Vol. 7, 2 (Feb. 1964), pp. 67-72.

N. Wirth, "Find Precedence Functions Algorithm 265", *Comm ACM*, Vol. 8, 10 (Oct. 1965), pp. 604-605.

N. Wirth, H. Weber, "Euler — A Generalization of ALGOL And Its Formal Definition." Part I, Part II., *Comm. ACM*, Vol. 9, 1 (Jan. 1966), pp. 13-23, 25 and Vol. 9, 2 (Feb. 1966), pp. 89-105, 107.

The first FORTRAN compiler required that an IBM 701 be adapted five distinct times to perform what are referred to as *passes* to produce suboptimally arranged operation codes for machine designers desiring to adapt the IBM 701 to their machine problem to be solved. This work commenced in 1952 and terminated in 1954. The technique was termed heuristic by W. Rutishauser, a word deriving from the German verb *heuriskein*, to invent by empirically determining alternatives. *Webster's New World Dictionary of The American Language*, Second College Edition (1972).

The evolution of a higher level language or, equivalently, translation algorithm, makes it clear that any such language or algorithm must have an efficient physical realization for carrying out an effective procedure for translation. The relationship of translation algorithm (higher level language) to translation effective procedure (implementation of the higher level language) is precisely the same as the relationship of an algorithm descriptive of an industrial context to an effective procedure for machine control within that industrial context. In fact, a *translation algorithm* must become a *translation effective procedure,* because in order that it be commercially viable as a machine, its functioning to carry a *source industrial algorithm* into its related *object industrial effective procedure* must be such that the operations defined for that object industrial effective procedure as well as other object structural definitions formed during compilation, e.g., shared facilities of the general purpose machine, must be more nearly optimal than competitive alternatives.

Typically, one utilizing the facilities of the enhanced or extended machine defined to compile his industrial algorithm over a period of time, and absent a requirement or the interest to "look in" to a greater depth, will avail himself of the best features of that machine. Thus, during the course of his preparation for carrying his industrial algorithm into an effective procedure for machine control, he is concerned with implementation, viz., getting the program machine which defines the effective procedure onto the general purpose machine to define a machine solution to the machine problem to be solved.

The extended/enhanced machine which is used for each and every step of compilation is a machine distinct from that which is to be structured by the fully-tested effective procedure. It is a machine for assisting a machine designer in the design of another machine. In the fully-formed machine, conceived as a distinct industrial algorithm and reduced to practice as a distinct industrial effective procedure for machine control, such control is achieved by hundreds of thousands of unique sequential interconnections of circuits. Thus, a general purpose machine, adapted as a compiler by an Academician responding to application pull, is more like an automatic machine for stamping printed circuits or an automatic lathe for machining parts than a translator of languages as asserted by petitioner (Pet. Br. 29).

The work of the Academician whose activities are within the second cycle, *supra* p. 18, may be described as axiomatic investigation, i.e., seeking a suitable axiomatic structure to define the physical structure of the various machines performing myriad complex computations to provide machine solutions within an ever-expanding industrial context. The search for appropriate techniques of axiomatic investigation is a continuing one. This type of investigation is of a unique character, in that it employs non-numerical mathematics.[26] This uniqueness is

---

[26]See Minsky, *supra* note 7, Preface, p. x:

"This really is a branch of mathematics that stems directly from non-numerical, logical foundations . . .

"It is worth a few words to try to explain why classical mathematical analysis plays such a small role here. In developing a Theory of Computation we are trying to deal with systems composed of a great many parts, or *very intricate structures* (emphasis added) . . . .Now it is true that under certain special conditions mathematical analysis can 'revive' as it were, when the situation gets complex in such a way that the parts of the system can be treated as individually and independently random . . . But it must be stated explicitly and emphatically that this is just what does *not* happen when, as in a computation system, the structure has a more organized, purposeful structure . . . There simply is no reason to suppose that as computations grow large, one will discover anything non-trivial by trying to 'average out' the effects of many events. The effect of the conditional is too strong . . .

"Fortunately, the systems of computation have other features that make possible some analysis, though a very different kind . . . we use logically defined classes of computations or expressions. They are tied together not by geometric or energetic properties, but by their relations to similar machines or similar definitions . . ."

reflected by the work of the many pioneers who have defined the nexus between the various techniques of axiomatic investigation.[27]

7. Summary

Invention in the program machine art may reside in conceptualizing a machine with the state of the implementation art in view, in reducing an abstraction to practice in the form of an implementation, or in any of the activities distributed over the various steps of the inventive process, according to the magnitude and complexity of the machine problem to be solved. The technological facts present a logical imperative, viz., that a new program defines a new machine which is either an extended or enhanced general purpose machine such as that of Amicus Universal or a special purpose machine such as that disclosed and claimed in respondent's application. Such new programs which meet the standards set by 35 U.S.C. §§102, 103 should be afforded patent protection[28] under that statute, particularly in view of the

[27]The following are exemplary of pioneer works defining adaptable theoretical structure employing the *axiomatic technique*, i.e., statements of propositions whose truth would be accepted within a framework of the logic of propositions respecting the description of machines not then in existence: Turing, *supra* note 5; S.C. Kleene, "General Recursive Functions of Natural Numbers," *Math Annalen*, 1936), pp. 112, 340-353; A. Church, "The Calculi of Lamda-Conversion", *Annals of Mathematics Studies*, Princeton (1941); E.L. Post, "Absolutely Unsolvable Problems And Relatively Undecidable Propositions — Account Of An Anticipation", M.S. unpublished, 1941, formally published, M. Davis, *The Undecidable*, Raven Press, Hewlett, N.Y. (1965); E.L. Post, "A Variant of A Recursively Unsolvable Problem," *Bull. Amer. Math. Soc.*, Vol. 52 (1946), pp. 264-268.

[28]Even the Solicitor General has publicly acknowledged that "the dynamic interrelationship between . . . computers and the software programs through which they are applied to numerous and varied tasks may call for new measures or legal concepts to further reward and stimulate innovation in programming." *Computerworld*, Vol. IX, No. 33 (August 13, 1975), p. 3.

unsuitability of both trade secret[29] and copyright protection.[30]

C. An Unobvious Idea Can Form A Sound Legal Basis For Patentability Even Though Its Implementation Is Obvious Once The Idea Has Been Grasped.

Petitioner has asserted that even where an idea[31] is not obvious, its implementation must be unobvious as well in order to justify the grant of Letters Patent, whether the implementation involves a process or a product (Pet. Br. 11-12, 22). Paraphrasing the foregoing proposition, petitioner has further asserted that the patent laws require not only the application of the unobvious idea to a new and useful end, but also the exercise of the inventive faculty in both conceiving and devising the end product or process (Pet. Br. 22-23). With these dangerously broad assertions, petitioner threatens a carefully-developed principle which has been firmly established in our patent law by a series of soundly-reasoned judicial decisions handed down over a period of nearly 75 years. This principle may be stated as follows: if a problem is unobvious or the source of a known problem is unobvious, and an inventor has recognized the existence of the unobvious problem or the unobvious source of the known problem, and has conceived and reduced to practice a new product (article, machine, composition of matter) or process which solves the problem, then that inventor may obtain patent protection for the new product or process, even if that product or process is an obvious solution to the unobvious problem after that problem has

[29]Scafetta, *supra* note 22, pp. 395-6. Also see *Computerworld*, *supra* note 28, p. 3, the comment by G. Chatfield on the near-impossibility of maintaining secrecy of proprietary software:

"When a main storage dump is taken, [the software vendor's] code lays in object form in that system. Anyone who is very knowledgeable knows you can do a fairly decent job in 'reverse translating' to get back to at least a pseudosource language. This more or less reveals the concepts, patterns and ideas of the vendor's software."

[30]Scafetta, *supra* note 22.

[31]"Idea" is defined as any conception existing in the mind as a result of mental understanding, awareness, or activity. *The Random House Dictionary of the English Language* (Unabridged Edition) (1966).

been recognized, or is an obvious solution to the known problem after its unobvious source has been recognized. *Hobbs v. Beach,* 180 U.S. 383, 392-3 (1901); *Eibel Process Company v. Minnesota & Ontario Paper Company,* 261 U.S. 45, 67-68 (1923); *General Electric Company v. Hartman,* 187 Fed. 131, 135-6 (1st Cir. 1911); *Application of Hamilton,* 64 F.2d 141,143 (C.C.P.A. 1933); *Application of Hofmann,* 95 F.2d 257, 259 (C.C.P.A. 1938); *Application of Bencker,* 96 F.2d 326, 328 (C.C.P.A. 1938); *Application of Deakins,* 96 F.2d 845, 849 (C.C.P.A. 1938); *Application of Earle,* 102 F.2d 232, 235 (C.C.P.A. 1939); *Application of DeLancey,* 159 F.2d 737, 741 (C.C.P.A. 1947); *Application of Emmey,* 161 F.2d 754, 756 (C.C.P.A. 1947); *Application of Holt,* 162 F.2d 472, 477 (C.C.P.A. 1947); *Application of Bisley,* 197 F.2d 355,363 (C.C.P.A. 1952); *Application of Tanczyn,* 202 F.2d 785, 787 (C.C.P.A. 1953); *Application of Shaffer,* 229 F.2d 476,480 (C.C.P.A. 1956); *Application of Pennington,* 241 F.2d 750, 754 (C.C.P.A. 1957); *Application of Hortman,* 264 F.2d 911, 913 (C.C.P.A. 1959); *Application of Antonson,* 272 F.2d 948, 949 (C.C.P.A. 1959); *Application of Conover,* 304 F.2d 680, 683-4 (C.C.P.A. 1962; *Application of Linnert,* 309 F.2d 498, 501-2 (C.C.P.A. 1962); *Application of Sponnoble,* 405 F.2d 578, 585-6 (C.C.P.A. 1969); *Trio Process Corp. v. L. Goldstein's Sons, Inc.,* 461 F.2d 66, 72 [footnote 18a] (3rd Cir. 1972), cert. denied 409 U.S. 997; *Application of Roberts,* 470 F.2d 1399, 1401 (C.C.P.A. 1973); *Application of Nomiya,* 509 F.2d 566, 571-2 (C.C.P.A. 1975). This principle has been applied not only in decisions by the Courts, but also in administrative decisions by the Patent Office. *Ex parte Phair,* 1 U.S.P.Q. 133, 134 (P.O. Bd. App. 1929); *Ex parte Greider,* 17 U.S.P.Q. 73 (P.O. Bd. App. 1932; *Ex parte Denny,* 65 U.S.P.Q. 523, 524 (P.O. Bd. App. 1944); *Ex parte Harr,* 73 U.S.P.Q. 120, 121 (P.O. Bd. App. 1946); *Ex parte Hemig,* 125 U.S.P.Q. 75, 76 (P.O. Bd. App. 1959); *Ex parte Ligh,* 159 U.S.P.Q. 61, 62 (P.O. Bd. App. 1967); *Ex parte Breidt,* 161 U.S.P.Q. 767, 768 (P.O. Bd. App. 1968). Thus, the

particular manifestation of invention[32] in which the idea or mental aspect is the crucial element from which the remaining elements obviously flow has consistently been found to be a patentable contribution to the useful arts.

The contrary proposition and the various paraphrasings thereof by petitioner are said to be precedented by *Gottschalk v. Benson, supra,* but the cited portion of that decision (67-68) speaks only of a requirement for the application of a newly-discovered law of nature to a new and useful (but not unobvious) end as a prerequisite to issuance of a patent based on the discovery. Thus, petitioner appears to have indulged in an unwarranted extrapolation of *Benson.* Also cited is *Funk Bros. Seed Co. v. Kalo Inoculant Co.,* 333 U.S. 127 (1948), in which product claims drawn to an inoculant for leguminous plants formed by combining mutually non-inhibitive strains of different species of bacteria were held invalid because they were found to be based on the newly-discovered law of nature that certain bacterial strains did not inhibit each others ability to fix nitrogen in specific leguminous

---

[32]Three classes of invention are defined in *Rosemary Mfg. Co. v. Halifax Cotton Mills,* 257 Fed. 321 (4th Cir. 1919) as follows:

"In many inventions there are two distinct steps: First, the conception of the general result wished for; second, the discovery of a way of obtaining it. In a large majority of cases, perhaps the first may be obvious to every one interested in a particular art, and it is the second which calls for the exercise of inventive genius. But that is not always so. It may well be that two or more machines, appliances, or tools are old and well known. Some day it dawns on someone that, if they are combined, new and useful results will be obtained. It may be that, so soon as the advantages of the combination are understood, the means of bringing it about are within the capacity of any fairly skilled mechanic. In a third class of cases inventive genius may be required both in perceiving the combination that is desirable, and in finding out a practical way of making it."

plants,[33] not because they were based on an idea. *Armour Pharmaceutical Co. v. Richardson-Merrell, Inc.*, 396 F.2d 70 (3rd Cir. 1968) dealt with product claims drawn to an enterically-coated oral dose of trypsin for systemic treatment of inflammation, the enteric coating being designed to protect the dose as it passed through the acid environment of the gastrointestinal tract until it reached the ileum, where it disintegrated permitting the release of the trypsin dose and its absorption into the bloodstream. Following *Funk*, and noting that the process of enterically coating medicaments was well-known, the Court held the product claims invalid because they were found to be based not on an idea, but on the discovery of the natural phenomenon that the ileum would absorb trypsin. *The Davison Chemical Corp. v. Joliet Chemicals, Inc.*, 179 F.2d 793 (7th Cir. 1950) involved patent claims to a process for making silica gel of low density (and therefore high porosity) in water of higher temperature than taught by the prior art. The utility of the silica gel as a dessicant was thereby enhanced in certain applications. The Court held the claims invalid, not because the invention derived from an idea, but because it was based on the newly-discovered scientific fact that the temperature of the washing water directly affects the density of the silica gel. In *National Lead Co. v. Western Lead Products Co.*, 324 F.2d 539 (9th Cir. 1963), the Court followed *Davison* in affirming the District Court decision invalidating claims to process for continuously produc-

[33]Mr. Justice Frankfurter, concurring in this result on the separate ground of inadequate disclosure, said:

"If this is a correct analysis of Bond's endeavors two different claims of originality are involved: (1) the idea that there are compatible strains, and (2) the experimental demonstration that there were in fact some compatible strains. Insofar as the court below concluded that the packaging of a particular mixture of compatible strains is an invention and as such patentable, I agree, provided not only that a new and useful property results from their combination, but also that the particular strains are identifiable and adequately identified."

Mr. Justice Burton and Mr. Justice Jackson, dissenting, agreed with Mr. Justice Frankfurter that the patentee's combinations of mutually non-inhibitive strains of bacteria satisfied the statutory requirements of invention.

ing a mixture of different types of lead, even though the process yielded superior control over the particle size, composition, and color of the mixture by restricting the mixing pot temperatures to a defined range. The Court found that the invention was based on the discovery of a natural relationship between the temperature of the reaction and the uniformity of product characteristics. In *Loew's Drive-In Theatres, Inc. v. Park-In Theatres, Inc.*, 174 F.2d 547 (1st Cir. 1949), cert. denied 338 U.S. 822, the structure of a drive-in theatre was acknowledged to be appropriate subject matter for a patent, but was found to be obvious as claimed, without giving any weight to the method of doing business in which that structure was intended to be instrumental.

Since only *Loew's* involved an invention which appears to be based upon an idea, and that relating to a method of doing business rather than a scientific or technical insight, petitioner's sweeping proposition is only narrowly supported by one case. It is noteworthy that the legal basis for the later decisions (except *Loew's*) derived largely from *Funk*, in which this Court was sharply divided.[34] In any event, all of these cases may be distinguished from those cited by Amicus Universal in that the discoveries in the latter cases were in a specific technological context, i.e., they were discoveries of deficiencies or problems or the sources of problems in the prior art, rather than discoveries of scientific principles *per se*. On this basis, it is urged that petitioner's excessively broad statement of what *Benson*, *Funk*, *Armour*, *National Lead*, *Davison*, and *Loew's* stand for be rejected so as to leave intact the soundly-reasoned body of law cited by Amicus Universal.

[34]Concerning the analysis in *Funk*, Mr. Justice Frankfurter, in his separate concurring opinion, *supra*, note 33, observed that:

"It only confuses the issue, however, to introduce such terms as 'the work of nature' and the 'laws of nature', for these are vague and malleable terms infected with too much ambiguity and equivocation. Everything that happens may be deemed 'the work of nature', and any patentable composite exemplifies in its properties 'the laws of nature'. Arguments drawn from such terms for ascertaining patentability could fairly be employed to challenge almost every patent."

To the extent that the few decisions cited by petitioner might be deemed to be inconsistent with the great weight of contrary authority cited by Amicus Universal, it is urged that the latter is more in keeping with the concept behind our patent system, viz., rewarding those inventors who make the knowledge of their technological advancements available to the public. An inventor who has made a contribution to the understanding of the useful arts should not be denied patent protection for an invention stemming from that understanding. To make an inquiry into the unobviousness of an invention without considering its conceptualization seems too narrow a view. Even if an invention might be readily carried out after forming its conceptualization, and even if that conceptualization begins with a newly-discovered scientific principle, the grant of a patent for a limited number of years does not seem too great a reward for the discovery and dissemination of a scientific principle which is eternal.

V

CONCLUSION

For the foregoing reasons, the judgement of the Court of Customs and Patent Appeals should be affirmed.

Respectfully submitted,

Charles Winn Sims
2208 Piedmont
Wichita Falls, Texas 76308

Counsel for Amicus Curiae

Of Counsel:

Francis Noel Carten
Carten & Sims
500 Summer Street
Stamford, Connecticut 06901

32 Broadway
Suite 514
New York, New York 10004